10th Class

Computer Science	Model Paper 4	Paper: II
Time: 1.45 Hours	(Subjective Type)	Marks: 40

(Part-I)

2. Write short answers to any FOUR (4) questions: (8)

(i) List down five reserved words in C programming language.

Ans Five reserved words in C programming language are:

1. auto 2. break 3. case
4. char 5. const

(ii) Write down the rules for naming variables.

Ans Each variable must have a unique name or identifier. Following rules are used to name a variable:

1. A variable name can only contain alphabets (uppercase or lowercase), digits and underscore_sign.
2. Variable name must begin with a letter or an underscore, it cannot begin with a digit.
3. A reserved word cannot be used as a variable name.
4. There is no strict rule on how long a variable name should be, but we should choose a concise length for variable name to follow good design practice.

(iii) What are escape sequences? Why do we need them?

Ans Escape sequences are used in *printf* function inside the "and". We need them because they force *printf* to change its normal behavior of showing output.

(iv) What is meant by precedence of operators? Which operator has the highest precedence in C language?

Ans If there are multiple operators in an expression, the question arises that which operator is evaluated first. An operator with higher precedence is evaluated before the operator with lower precedence. In case of equal precedence, the operator at left side is evaluated before the operator at right side. ( ) operator has the highest precedence in C language.

(v) Write down output of the following code:

```
# includes<stdio.h>
void main ()
{
        printf ( “nn \n\n nnn\nn\nt\t” ) ;
        printf ( “nn /n/n nn/n\n” ) ;
}
```

Ans Output:

```
nn
nnn
n
t      nn/n/n nn/n
```

(vi) Identify errors in the following code:

```
#include<stdio.h>;
main ()
{
        int num;
        printf(Enter number: ");
        scanf(%d, &num);
};
```

Ans Error:

1. Semicolon is extra at the end of Header file.
2. Inverted comma is missing in scanf statement.
3. Semicolon is not used after the end of main function curly braces.
4. # is missing at the start of header file.
5. printf is a function so there is use parenthesis () instead of square brackets [ ].

3. Write short answers to any FOUR (4) questions: (8)

(i) Write the structure of *if statement.*

Ans **Structure of if statement:**

If statement has the following structure in C language:

```
if (condition)
    Associated Code
```

(ii) Identify error in the following code:

```
if (a == b & | x == y)
        flag = true;
else
        flag = false;
```

Ans Error: OR (||) operator is used as double pipe line but single is used.

(iii) **Write down output of the following code:**

```
int a = 34, b = 32, c = 7, d = 15;
a = b + c + d;
if (a < 100)
        a = a * 2;
b = b * c;
c = c + d;
if (a > b && c == d)
{
        c = d;
        b = c;
        a = b;
}
else
        if (a > b && c > d || b >= b + c)
        {
                d = c * c;
                a = b * c;
        }
printf ("a=%d, b = %d, c = %d, d = %d", a, b, c, d);
```

Ans Output: a = 50176, b = 224, c = 22, d = 484

(iv) **Define the loop structure.**

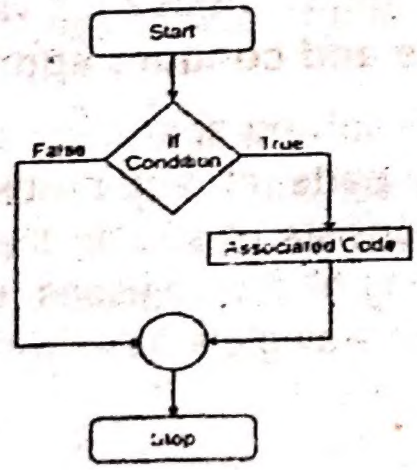
Ans **Loops structure** is used to repeat a set of statements. Three types of loops are for loop, while loop, do-while loop.

(v) **Define the nested loops.**

Ans We use **nested loops** to repeat a pattern multiple times.

(vi) **Draw a flow chart to show the basic flow of an if statement.**

Ans



4. Write short answers to any FOUR (4) questions: (8)

(i) Identify the error in the following code:

```
float f[ ] = {1.4, 3.5, 7.3, 5.9};
int size = 4;
for (int n = –1; n < size; n --)
        printf (“%f\n”, f[n]);
```

Ans Loop condition cannot be wrong.

(ii) Write down output of the following code:

```
int i, arr[ ] = {2, 3, 4, 5, 6, 7, 8};
for (i = 0; i < 7; i++)
{
        printf (“%d\n”, arr [i] * arr[i]);
        i++;
}
```

Ans **Output:**

```
     4
     9
     16
     25
```

(iii) What is ‘Calling a function’?

Ans **Calling a function** means to transfer the control to that particular function.

(iv) Is it necessary to use compatible data types in function definition and function call? Justify your answer with an example.

Ans **Using a function:**

We need to call a function, so that it performs the programmed task. Following is the general structure used to make a function call.

function_name(value1, value2, ..., valueN);

(v) What is divide and conquer approach?

Ans A good problem solving approach is to divide the problem into multiple smaller parts of sub-problems. Solution of the whole problem thus consists of solving the sub-problems one by one, and then integrating all the solutions. In this way, it becomes easier for us to focus on a single smaller problem at a time, instead of thinking about the whole problem all the time. This problem solving approach is called divide and conquer approach.

(vi) What is the advantage of handling the complexity of the problem?

Ans If we write the whole program as a single procedure, management of the program becomes difficult. Functions divide the program into smaller units, and thus reduce the complexity of the problem.

(Part-II)

NOTE: Attempt any TWO (2) questions.

Q.5. What is variable declaration? Give some examples. (8)

Ans **Variable Declaration:**

We need to declare a variable before we can use it in the program. Declaring a variable includes specifying its data type and giving it a valid name. Following syntax can be followed to declare a variable.

```
data_type variable_name;
```

Examples:

Some examples of valid variable declarations are as follows:

```
unsigned int age;
float height;
int salary;
char marital_status;
```

Multiple variables of same data type may also be leclared in a single statement, as shown in the following examples:

```
unsigned int age, basic_salary, gross_salary;
int points_scored, steps;
float height, marks;
char marital_status, gender;
```

A variable cannot be declared unless we mention its data ype. After declaring a variable, its data type cannot be hanged. Declaring a variable specifies the type of variable, the ange of values allowed by that variable, and the kind of perations that can be performed on it. Following example hows a program declaring two variables:

```
void main ( )
{
        char grade;
        int value;
    }
```

Q.6. Write a program that calculates the volume of cube, cylinder or sphere, according to the choice of user. (8)

Ans

```
#include<stdio.h>
void main ()
{
        int choice;
        float volume;
        printf ("Find Volume\n");
        printf  ("1.Cube\n2.Cylinder\n3.Sphere\nEnter  your
        choice :");
        scanf ("%d", &choice);
        if (choice == 1)
        {
            float length;
            printf ("Enter Length: ");
            scanf("%f", &length);
            volume = length * length * length;
            printf ("Volume is %f", volume);
        }
        else if (choice == 2)
        {
            float length1, radius1;
            printf ("Enter Length: ");
            scanf ("%f", & length1);
            printf ("Enter Radius: ");
            scanf ("%f", & radius 1);
            volume = 3.142 * radius1 * radius1 * length1;
            printf ("Volume is %f", volume);
        }
        else if (choice == 3)
        {
            float radius;
            printf ("Enter Radius: ");
            scanf ("%f", & radius);
            volume = 3.142 * radius * radius * radius;
            printf ("Volume is %f", volume);
        }
        else
            printf ("Invalid Choice");
}
```

Q.7. What do you know about nested loops? When do we use nested loops? (8)

Ans **Nested Loops:**

Let's carefully observe the general structure of a loop.

```
for(initialization; condition; increment / decrement)
{
        Code to repeat
}
```

We can observe that Code to repeat could be any valid C language code. It can also be another for loop *e.g.,* the following structure is a valid loop structure.

```
for (initialization; condition; increment / decrement)
{
        for (initialization; condition; increment / decrement)
        {
            Code to repeat
        }
}
```

When we use a loop inside another loop, it is called nested loop structure.

When do we use nested loops?

When we want to repeat a pattern for multiple times, then we use nested loops, *e.g.,* if 10 times we want to display the numbers from 1 – 10. We can do this by writing the code of displaying the numbers from 1 – 10 in another loop that runs 10 times.